

一、 名词解释（每小题 2 分）

1. 地址映射

为了保证 CPU 执行指令时可正确访问存储单元，需将用户程序中的逻辑地址转换为运行时

由机器直接寻址的物理地址

2. 动态重定位

动态重定位的目的是为了让程序能够在不同的内存地址空间中运行，从而避免内存地址的硬

编码

3. 虚拟存储器

虚拟存储器是通过内存管理技术，使得应用程序能够使用比物理内存更大的地址空间

4. 静态链接

在编译程序时将程序所依赖的库文件的代码和数据复制到可执行文件中，生成一个完全独立

的可执行文件

5. 对换

把内存中暂时不能运行的进程或者暂时不用的程序和数据换出到外存上

6. 设备驱动程序

可以使计算机和设备通信的特殊程序

7. SPooling

同时联机外围操作技术，它是关于慢速字符设备如何与计算机主机进行数据交换的一种技术

8. I/O 通道

I/O 通道是用于处理 I/O 操作的硬件或软件，可以独立于 CPU 进行 I/O 操作

9. 文件系统

操作系统用于明确存储设备（常见的是磁盘，也有基于 NAND Flash 的固态硬盘）或分区上

的文件的方法和数据结构

10. 目标文件

编译器编译源代码后生成的文件

11. 文件的逻辑结构

文件的逻辑结构是指文件从用户角度看起来的组织形式，如记录的顺序

12. 有结构文件

有结构文件是指文件内部有特定的组织结构，如数据库文件。

13. 位示图

利用二进制的一位来表示磁盘中一个盘块的使用情况

14. 程序接口

操作系统提供给应用程序的一组调用方法，用于请求操作系统服务

15. 系统调用

操作系统提供给应用程序使用的接口

16. I/O 中断

I/O 中断通过中断处理器执行中断操作

17. 文件管理系统

负责管理和存储文件信息的软件机构

18. 文件

操作系统中的文件是一种抽象机制，它提供了一种在磁盘上保存信息而且方便以后读取的方法

19. 文件的逻辑结构

指文件在用户视角下呈现的组织形式

20. 文件的物理结构

文件在外存上的存储组织形式

二、 填空题（每空 1 分）

1. 程序被装入内存时由操作系统的连接装入程序完成程序的逻辑地址到内存地址的转换，也称为_____动态重定位_____。

2. 程序的装入方式包括_____绝对装入_____、_____运行时动态装入_____和_____可重定位装入_____三种。

3. 程序的链接方式包括_____静态链接_____、_____装入时动态链接_____和_____运行时动态链接_____三种。

4. 在分区管理方式中，空闲分区的管理所使用的数据结构包括_____空闲分区表_____和_____空闲分区链_____。

5. 将系统中所有空闲的小分区集中起来形成一个大分区的过程称为_____紧凑_____。

6. 比较分页与分段管理，页的大小是_____固定_____的，由_____系统_____决定，而段的大小是_____不固定_____的，由_____用户_____决定。

7. 虚拟存储器的主要特征包括多次性、_____对换性_____和_____虚拟性_____。

8. 文件系统由_____文件管理器_____、_____目录管理器_____和_____存储空间管理器_____三部分组成。

9. 内存的动态分区分配方式涉及：_____数据结构_____、_____分配算法_____、_____分配操作_____这三方面的问题。

10. 内存连续分配方式容易产生_____内存碎片____, 从而降低内存的使用率。
11. 内存的离散分配方式大致可分为: _____分页____方式、
_____分段____方式、_____段页____方式。
12. 在内存的离散分配管理方式下, 为了提高地址变换速度, 可增设一种特殊的缓冲寄存器, 称为_____快表_____。
13. 分页存储管理将进程的_____逻辑地址____和_____内存____空间分为相同大小的页面。
14. 分段存储管理方式的优点包括: _____高效性____、_____便捷性____、
_____可扩展____、

共享与保护、_____模块化_____。

15. 地址映射机构实现进程从逻辑地址到物理地址的变换功能。
16. 程序运行时存对内存访问存在局部性现象, 表现为_____空间局部性____和时间局部性。
17. I/O通道是一种特殊的处理机, 但与其差别在于(1) _____指令集过少_____(2) 无分配内存
18. 有结构文件的组织方式可大致分为_____索引文件____、_____顺序文件____和
_____索引顺序文件____
三大类。
19. 典型的磁盘索引组织方式包括_____单级索引____、_____多级索引____和
_____。
20. 在作业执行期间才进行的地址变换方式是_____动态重定位_____。
21. I/O管理系统软件可以分为三个层次: _____设备驱动程序____、_____中断
处理程序____、_____IO控制器_____。
22. 在_____顺序____、_____索引____和_____直接____三种文件的物理结构中,
随机访问效率最高的是_____直接文件_____。
23. 计算机系统存储器的存储大致可分为_____外存____、_____主存____和_____高
速缓冲存储器____三个层次。
24. 计算机主存分配存储管理方式可分为_____连续存储____和_____离散存储____
方式。
25. 文件的物理结构不仅与_____存储介质____有关, 而且与_____访问方式____有
关。
26. 根据记录的组织方式, 可把文件的逻辑结构分为_____索引____、_____顺序____
和_____索引顺序____三大类。
27. 外存的分配方式有_____连续____、_____链式____和_____索引____三大类。
28. 在树型目录结构中, 根据路径的起点不同, 可把路径分为_____相对路径____和
_____绝对路径____两种。
29. 在采用空闲链表法来管理空闲盘区时, 有_____显式链表____和_____隐式链表____
两种形式。

30. 文件的共享分为_____显式共享_____和_____隐式共享_____两种方式。
31. 按照信息交换的单位可把设备分为_____字符设备_____和_____块设备_____两大类；而按照设备的共享属性又可把设备分为_____独占设备_____、_____共享设备_____和_____虚拟设备_____三大类。
32. I/O 设备的控制方式可分为程序控制方式、_____DMA方式_____、中断方式和_____通道方式_____等。
33. 为了缓冲 CPU 与 I/O 设备速度不匹配的矛盾，在CPU 和 I/O 设备之间引入了缓冲技术，缓冲可分为_____单缓冲_____、双缓冲、_____和_____高速缓冲_____四种。
34. 磁盘访问时间包括_____寻道时间_____、旋转延迟时间和_____传输时间_____。

三、 简答题（每小题 6 分）

1. 计算机存储器系统主要有哪些层次？各个层次又包含哪些内容？试从存储性能角度对存储系统进行分析。

计算机中的存储器系统层次结构通常包括寄存器、高速缓存、内存和外部存储器。

寄存器（Registers）：

内容：CPU 内部的高速存储单元，用于存储指令、数据和地址等。

性能：访问速度极快，但容量非常有限。

缓存（Cache）：

内容：位于 CPU 和主存之间的高速存储器，用于暂存 CPU 频繁访问的数据和指令。

性能：访问速度接近寄存器，容量大于寄存器但小于主存。

内存（Main Memory 或 Random Access Memory, RAM）：

内容：内存是计算机的主要工作存储器，用于存储当前运行的程序和数据。

性能：访问速度较快，容量较大，但比高速缓存慢。

外部存储器（Secondary Storage 或 Auxiliary Memory）：

内容：包括硬盘驱动器（HDD）、固态硬盘（SSD）、光盘、USB 闪存等，用于长期存储数据和程序。

性能：访问速度较慢，容量非常大，成本相对较低。

2. 简述计算机程序从代码编写到运行完毕所经历的主要过程。

阶段一：预处理（Preprocessing）

在这个阶段，预处理器将源代码中的预处理指令（如 `#include`、`#define` 等）替换为实际

的内容。预处理器会根据指令展开头文件，处理宏定义，并删除注释等。

阶段二：编译（Compilation）

编译器将预处理后的源代码转换为汇编代码。它会对每个源文件进行词法分析、语法分析

和语义分析，生成相应的中间代码，。

阶段三：汇编（Assembly）

汇编器将编译生成的中间代码转换为机器可读的汇编代码，也就是机器码的文本表示。这

个阶段将源代码转换为机器语言的汇编指令，但还没有生成可执行文件。

阶段四：链接（Linking）

链接器将汇编生成的目标文件（object file）以及可能的库文件链接在一起，生成最终的可

执行文件。它会解析各个目标文件之间的符号引用和符号定义关系，将它们合并成一个完

整的程序。

3. 程序链接的主要方式有哪些？它们的异同点是什么？

静态链接

在程序运行前，将目标模块及所需的库函数连接成一个完整的装配模块，以后不在拆开

装入时动态链接

指将用户源程序编译后所得的一组目标模块，在装入内存时，采用边装入边链接的方式

运行时动态链接

指对某些目标模块的链接，是在程序执行中需要该目标模块时，才对它进行链接

4. 内存动态分区分配算法根据搜索空闲区的方式可分为哪些类型，它们又分别有哪些典型的分配算法？请逐一简述这些算法的核心思想。

首次适应算法（First Fit）：从内存的起始位置开始查找，找到第一个能够满足进程需要的空闲分区进行分配。

最佳适应算法（Best Fit）：从所有空闲分区中找到最小的能够满足进程需要的空闲分区进行分配。

最坏适应算法（Worst Fit）：从所有空闲分区中找到最大的能够满足进程需要的空闲分区进行分配。

快速适应算法（Quick Fit）：将内存分为若干个大小相等的分区，每个分区维护一个空闲链表，根据进程需要的大小在相应的链表中查找空闲分区进行分配。

列举典型的内存动态分区算法和 CPU 调度算法，并分析两者之间的异同点。

内存动态分区算法：

首次适应算法（First Fit）：

从内存的起始位置开始搜索，分配第一个足够大的空闲分区。

最佳适应算法（Best Fit）：

搜索整个内存，找到能够满足需求且大小最小的分区。

最坏适应算法（Worst Fit）：

找到最大的分区，即使它比需求大得多。

循环首次适应算法 (Next Fit) :

类似首次适应, 但每次搜索都是从上上次搜索结束的位置开始。

CPU 调度算法:

先来先服务 (FCFS, First-Come, First-Served) :

按照进程到达的顺序进行调度。

短作业优先 (SJF, Shortest Job First) :

优先调度估计执行时间最短的进程。

优先级调度 (Priority Scheduling) :

根据进程的优先级进行调度, 优先级高的进程先执行。

时间片轮转 (RR, Round Robin) :

每个进程被分配一个固定的时间片, 按顺序轮流执行。

异同点分析:

资源分配:

内存分区算法分配的是内存空间, 而 CPU 调度算法分配的是 CPU 时间。

目标:

内存分区算法的目标是有效利用内存空间, 减少外部碎片。

CPU 调度算法的目标是公平、高效地分配 CPU 时间, 提高系统吞吐量。

性能影响:

内存分区算法影响程序的加载时间和内存利用率。

CPU 调度算法影响进程的响应时间和等待时间。

算法选择:

内存分区算法的选择取决于系统对内存利用率和分配速度的需求。

CPU 调度算法的选择取决于系统对响应时间和公平性的需求。

碎片问题:

内存分区算法可能导致内部碎片或外部碎片。

CPU 调度算法可能导致 CPU 时间的碎片化, 尤其是在时间片轮转算法中。

动态性:

动态内存分区算法需要在运行时根据需求动态调整内存分配。

CPU 调度算法通常也是动态的, 需要根据进程的状态和优先级进行调整。

算法复杂度:

内存分区算法的复杂度可能因算法不同而异, 如最佳适应算法可能需要较高的搜索成本。

CPU 调度算法的复杂度通常较低, 但优先级调度可能需要维护优先级队列。

5. 列举典型的页面置换算法和 CPU 调度算法, 并分析两者之间的异同点。

一、页面置换算法 (Page Replacement Algorithms)

用于虚拟存储器管理, 当页面缺页而内存已满时, 系统需要选择一页淘汰, 以换入所需页面。

典型算法包括:

FIFO (先进先出) 算法

选择最早进入内存的页面置换。

LRU (最近最少使用) 算法

置换最长时间未被访问的页面

OPT（最佳置换）算法

置换未来最长时间不会被访问的页面（理想状态，用于比较）。

Clock（时钟）算法

一种近似LRU的算法，使用环形队列和访问位。

二、CPU 调度算法（CPU Scheduling Algorithms）

用于多道程序设计环境中，决定哪个进程获得 CPU 使用权，提高系统资源利用率与响应时间。

典型算法包括：

FCFS（先来先服务）

按到达顺序调度，不可抢占。

SJF（短作业优先）

优先执行运行时间短的作业，通常不可抢占。

RR（时间片轮转）

每个进程按时间片轮流执行，适用于分时系统。

优先级调度

按进程优先级分配 CPU。

多级反馈队列

结合优先级和时间片，综合性能强。

三、异同点分析

资源分配：

内存分区算法分配的是内存空间，而 CPU 调度算法分配的是 CPU 时间。

目标：

内存分区算法的目标是有效利用内存空间，减少外部碎片。

CPU 调度算法的目标是公平、高效地分配 CPU 时间，提高系统吞吐量。

性能影响：

内存分区算法影响程序的加载时间和内存利用率。

CPU 调度算法影响进程的响应时间和等待时间。

算法选择：

内存分区算法的选择取决于系统对内存利用率和分配速度的需求。

CPU 调度算法的选择取决于系统对响应时间和公平性的需求。

碎片问题：

内存分区算法可能导致内部碎片或外部碎片。

CPU 调度算法可能导致 CPU 时间的碎片化，尤其是在时间片轮转算法中。

动态性：

动态内存分区算法需要在运行时根据需求动态调整内存分配。

CPU 调度算法通常也是动态的，需要根据进程的状态和优先级进行调整。

算法复杂度：

内存分区算法的复杂度可能因算法不同而异，如最佳适应算法可能需要较高的搜索成本。

CPU 调度算法的复杂度通常较低，但优先级调度可能需要维护优先级队列。

6. 列举典型的页面置换算法和磁盘调度算法，并分析两者之间的异同点。

页面置换算法：

最近最少使用 (LRU, Least Recently Used) :

淘汰最长时间未被访问的页面。

先进先出 (FIFO, First In First Out) :

按照页面进入内存的顺序进行淘汰。

最佳置换 (OPT, Optimal Replacement) :

理想情况下, 淘汰未来最长时间不会被访问的页面。

时钟置换 (Clock) :

使用一个类似时钟的数据结构, 按顺序检查页面, 淘汰最老的页面。

随机置换 (Random) :

随机选择一个页面进行淘汰。

磁盘调度算法:

先来先服务 (FCFS, First-Come, First-Served) :

按照请求到达的顺序进行服务。

最短寻道时间优先 (SSTF, Shortest Seek Time First) :

优先服务距离当前磁头位置最近的请求。

扫描 (SCAN) :

磁头在磁盘两端之间来回扫描, 每次移动到离当前位置最近的请求。

循环扫描 (C-SCAN) :

类似扫描算法, 但磁头在完成一次扫描后直接移动到另一端, 而不是返回。

最近最少使用 (NRU, Not Recently Used) :

类似内存中的 LRU 算法, 优先服务最近最少被访问的请求。

异同点分析:

资源类型:

页面置换算法管理的是内存中的页面。

磁盘调度算法管理的是磁盘 I/O 请求。

目标:

页面置换算法的目标是减少页面错误, 提高内存利用率。

磁盘调度算法的目标是减少寻道时间和旋转延迟, 提高磁盘访问效率。

性能影响:

页面置换算法影响程序的执行效率和响应时间。

磁盘调度算法影响 I/O 请求的响应时间和系统吞吐量。

算法复杂度:

页面置换算法中, OPT 算法具有最高的理论性能, 但实现复杂度也最高。

磁盘调度算法中, SSTF 算法可能导致饥饿现象, 需要额外的策略来平衡。

实现方式:

页面置换算法通常在内存管理单元中实现。

磁盘调度算法在磁盘控制器或 I/O 子系统中实现。

数据结构:

页面置换算法可能使用队列、栈、链表等数据结构。

磁盘调度算法可能使用队列、双端队列等数据结构。

动态性:

页面置换算法和磁盘调度算法都需要动态地根据当前状态做出决策。

公平性:

页面置换算法通常不需要考虑公平性问题。

磁盘调度算法需要考虑请求的公平性, 避免某些请求被饿死。

7. 分别简述分页存储、分段存储和段页存储的地址变换过程。

分页存储 (Paging) :

地址变换过程首先提取虚拟地址中的页号。
页号用于索引页表，找到对应的物理页号。
物理页号与页内偏移组合形成完整的物理地址。

分段存储 (Segmentation) :

地址变换过程首先提取虚拟地址中的段号。
段号用于索引段表，找到对应的基地址和段长度。
基地址加上段内偏移形成完整的物理地址。
在访问之前，还需要检查段内偏移是否超出段长度，以防止越界访问。

段页存储 (Segmentation with Paging) :

地址变换过程首先提取虚拟地址中的段号。
段号用于索引段表，找到对应的页表基地址和段属性。
段属性可能包括页的大小和页表长度。
将页号与页表基地址组合，形成页表的物理地址。

在页表中索引页号，找到对应的物理页号。
物理页号与页内偏移组合形成最终的物理地址。
同样需要检查页内偏移是否超出页的大小。

8. 简述中断处理程序的各个处理步骤。

1. 中断请求

当硬件设备或软件需要 CPU 注意时，它会发出中断请求信号。

2. 中断优先级判定 (简称中断判优)

操作系统根据中断请求的优先级来决定是否立即响应该中断。如果当前正在处理更高优先级的中断，低优先级的中断请求可能需要等待。

3. 中断响应

一旦 CPU 决定响应中断请求，它会进入中断响应阶段，这通常涉及到硬件和软件的协作，以确保 CPU 能够正确地处理即将到来的中断。

4. 中断处理

保存当前任务的上下文，如寄存器状态和程序计数器。
识别中断源，可能通过查询中断向量表。
执行中断服务，如数据的读取、硬件状态的更新、错误处理等。

5. 中断返回

恢复之前保存的上下文，使 CPU 能够从中断发生的位置继续执行。
如果需要，重新开启中断，允许新的中断请求被处理。
可能涉及调度其他任务，如果中断处理改变了任务的优先级或状态。

9. 从用途、数据类型、组织和管理方式等角度简述文件类型分类。

1) 按用途分类

根据文件的性质和用途的不同，可将文件分为三类：

系统文件。由系统软件构成的文件。大多数的系统文件只允许用户调用，但不允许用户去读，更不允许修改。

用户文件。由用户的源代码、目标文件、可执行文件或数据等所构成的文件。用户将这些文件委托给系统保管。

库文件。由标准子例程及常用的例程等所构成的文件。这类文件允许用户调用，但不允许修改。

2) 按文件中数据的形式分类

源文件。由源程序和数据构成的文件。通常由终端或输入设备输入的源程序和数据所形成的文件都属于源文件。

目标文件。把源程序经过相应语言的编译程序编译过，但尚未经过链接程序链接的目标代码所构成的文件。它属于二进制文件。

通常，目标文件所使用的后缀名是".obj".

可执行文件。把编译后所产生的目标代码再经过链接程序链接后所形成的文件。

3) 按存取控制属性分类

根据系统管理员或用户所规定的存取控制属性，可将文件分为三类：

只执行文件。只允许被核准的用户调用执行，既不允许读，更不允许写。

只读文件。只允许文件主及被核准的用户去读，但不允许写。

读写文件。允许文件主和被核准的用户去读或写的文件。

4) 按组织形式和处理方式分类

根据文件的组织形式和系统对其的处理方式，可将文件分为三类：

普通文件。由 ASCII 码或二进制码组成的字符文件。一般用户建立的源程序文件、数据文件、目标代码文件及操作系统自身代码文件、库文件等都是普通文件，它们通常存储在外存储设备上。

目录文件。由文件目录组成的，用来管理和实现文件系统功能的系统文件，通过目录文件可以对其它文件的信息进行检索。由于目录文件也是由字符序列构成，因此对其可进行与普通文件一样的种种文件操作。

特殊文件。特指系统中的各类 I/O 设备。为了便于统一管理，系统将所有的输入/输出设备都视为文件，按文件方式提供给用户使用

10. 请详述文件目录的分类及相应查询方式。

一级目录结构

遍历检索

二级目录结构

树结构：根结点主目录、分支结点用户目录、子分支结点文件

多级目录文件

必须指出文件所在路径名：从根目录到该文件的路径上各级目录名的组合。也称为文件全名。

11. 请简述文件结构的三种主要组织方式，并对比分析各自优劣。

一. 顺序结构，

优点

- 1、简单：存储与管理都简单，且容易实现。
- 2、支持顺序存取和随机存取。
- 3、顺序存取速度快。
- 4、所需的磁盘寻道次数和寻道时间最少。

缺点

- 1、需要为每个文件预留若干物理块以满足文件增长的部分需要。
- 2、不利于文件插入和删除。

二. 链式结构

优点

- 1、提高了磁盘空间利用率，不需要为每个文件预留物理块。
- 2、有利于文件插入和删除。
- 3、有利于文件动态扩充。

缺点

- 1、存取速度慢，不适于随机存取。
- 2、当物理块间的连接指针出错时，数据丢失。
- 3、更多的寻道次数和寻道时间。
- 4、链接指针占用一定的空间，降低了空间利用率。

三. 索引结构

优点

- 1、不需要为每个文件预留物理块。
- 2、既能顺序存取，又能随机存取。
- 3、满足了文件动态增长、插入删除的要求。

缺点

- 1、较多的寻道次数和寻道时间。
- 2、索引表本身带来了系统开销。如：内外存空间，存取时间等

12. 请分别简述内存和外存的存储分配空间分配方式，并对比分析它们之间的异同点。

内存的存储分配方式：

1.静态分配：

在程序编译时确定内存分配，通常用于操作系统内核和一些关键的系统程序。

2. 动态分配：

在程序运行时根据需要分配和释放内存，如 C 语言中的`malloc`和`free`。

3. 分页：

内存被分为固定大小的页，每个页可以独立加载到内存中。

4. 分段：

内存根据程序的逻辑结构被分为不同的段，如代码段、数据段等。

5. 段页式：

结合了分页和分段的特点，提供了更灵活的内存管理。

外存的存储分配方式：

1. 连续分配：

为文件分配连续的存储空间，如旧的文件系统。

2. 链式分配：

文件的每个块可以分散存储，通过链表结构连接。

3. 索引分配：

使用一个索引块来记录文件块的位置，适用于大型文件。

异同点分析：

相同点：

两者都需要有效的空间管理策略，以确保存储资源的合理利用。

都可能使用索引或表来跟踪存储空间的使用情况。

都面临着如何高效分配和回收空间的问题。

不同点：

访问速度：内存的访问速度远高于外存，因此内存的分配方式更注重快速访问和上下文切换。

存储介质：内存通常使用 RAM，是易失性的；外存使用硬盘、固态硬盘等，是非易失性的。

使用场景：内存分配主要关注当前运行的程序和数据，而外存分配需要考虑长期存储和数据备份。

13. 请分别简述提高磁盘 I/O 速度的多种途径。

提高磁盘 I/O 速度的途径：采用磁盘高速缓存是目前提高磁盘 I/O 速度的主要技术。

提前读：对于采用顺序访问方式访问文件，可提前将下一盘块的数据读入高速缓存。

延迟写：将可能会被其他进程访问的数据挂到空闲缓冲队列的末尾，而不是写回磁盘。

优化物理块的分布：将一个文件尽量存储在相邻的盘块上，以减少磁头切换磁道。

使用更快的磁盘